

MICHAEL A. JACOBS (CA SBN 111664)
MJacobs@mofo.com
DIEK O. VAN NORT (CA SBN 273823)
DVanNort@mofo.com
MORRISON & FOERSTER LLP
425 Market Street
San Francisco, California 94105-2482
Telephone: (415) 268-7000
Facsimile: (415) 268-7522

ROSE S. LEE (CA SBN 294658)
RoseLee@mofo.com
MORRISON & FOERSTER LLP
707 Wilshire Boulevard, Suite 6000
Los Angeles, California 90017-3543
Telephone: (213) 892-5200
Facsimile: (213) 892-5454

KYLE W.K. MOONEY (*Pro Hac Vice*)
KMooney@mofo.com
ERIC W. LIN (*Pro Hac Vice*)
ELin@mofo.com
MICHAEL J. DESTEFANO (*Pro Hac Vice*)
MDeStefano@mofo.com
MORRISON & FOERSTER LLP
250 West 55th Street
New York, New York 10019-9601
Telephone: (212) 468-8000
Facsimile: (212) 468-7900

Attorneys for Defendant
PALO ALTO NETWORKS, INC.

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

FINJAN LLC,

Plaintiff,

v.

PALO ALTO NETWORKS, INC.,

Defendant.

Case No. 3:14-CV-04908-JD

**PAN'S REPLY IN SUPPORT OF
MOTION TO STRIKE FINJAN'S
INFRINGEMENT
CONTENTIONS FOR THE '154,
'408, AND '731 PATENTS AND
TO DIMISS THESE PATENT
CLAIMS WITH PREJUDICE**

**[REDACTED VERSION OF
DOCUMENT SOUGHT TO BE
SEALED]**

Date: November 17, 2022
Time: 10:00 a.m.
Courtroom: 11, 19th Floor
Judge: Honorable James Donato

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. ARGUMENTS	2
A. Finjan’s Contentions for the ’154 Patent Are Deficient.....	2
1. “input” and “content”	2
2. “first function” and “second function”	4
3. “content processor” and “security computer”	5
B. Finjan’s Contentions for the ’408 Patent Are Deficient.....	6
1. “parser rules” and “analyzer rules”	6
2. “scanner”	7
C. Finjan’s Contentions for the ’731 Patent Are Deficient.....	8
1. “file cache” and “security profile cache”	8
2. “security policy cache”	10
III. CONCLUSION	10

TABLE OF AUTHORITIES**Page(s)****Cases**

<i>Finjan, Inc. v. Check Point Software Techs., Inc.</i> , No. 18-CV-02621-WHO, 2019 WL 955000 (N.D. Cal. Feb. 27, 2019).....	2
<i>Finjan, Inc. v. FireEye, Inc.</i> , No. 13-cv-03133-SBA (JCS), Dkt. No. 134 (N.D. Cal. Oct. 16, 2017).....	9
<i>Network Caching Tech. LLC v. Novell Inc.</i> , No. C-01-2079-VRW, 2002 WL 32126128 (N.D. Cal. Aug. 13, 2002).....	7
<i>Shared Memory Graphics LLC v. Apple, Inc.</i> , 812 F. Supp. 2d 1022 (N.D. Cal. 2010)	9
<i>Uniloc USA, Inc. v. Apple Inc.</i> , No. C 18-00360 WHA, 2018 WL 3219486 (N.D. Cal. July 2, 2018)	10
<i>Word to Info Inc. v. Google Inc.</i> , No. 15-CV-03486-WHO, 2016 WL 3648605 (N.D. Cal. July 8, 2016).....	8

Other Authorities

N.D. Cal. Patent L.R. 3-1(c)	9
------------------------------------	---

I. INTRODUCTION

Finjan is well aware of the level of detail required to satisfy the Patent Local Rules. Both this Court and Judge Hamilton have previously ordered Finjan to serve compliant infringement contentions. (Dkt. Nos. 146, 177.) And in at least seven other cases, courts in this district have found that Finjan violated the Patent Local Rules by serving deficient infringement contentions. (See Dkt. No. 128 at 2 n.1.) Yet Finjan’s Opposition confirms that its Second Amended Contentions again fail to identify *where* and *how* the asserted claim limitations are found in PAN’s products. While they parrot the claim language in some spots or make conclusory statements about the claim limitations being met in other spots, Finjan’s contentions overall fail to follow the claim language and are largely empty of the required substance. Finjan claims that reading its contentions “in total” or in “full view” will allow one to discover Finjan’s infringement theories. But even when read in full, Finjan’s contentions do not provide the required where and how.

Finjan also misstates the case history and tries to shift the blame for PAN being forced to file multiple motions to strike (the first two of which were granted) to obtain Finjan’s compliance with the Patent Local Rules. As early as March 11, 2021, PAN reminded Finjan of its obligations to serve compliant infringement contentions.¹ (Dkt. No. 104 at 23-25.) But after three rounds of infringement contentions, Finjan has still not done so. A month after Finjan served its Second Amended Contentions, PAN wrote Finjan detailing the deficiencies of Finjan’s contentions. (Dkt. No. 200-2.) The parties subsequently conferred, but Finjan refused to correct the deficiencies. (Dkt. No. 199-3.) This case was then in “a *de facto* stay,” as Finjan acknowledged in its scheduling order briefing. (Dkt. No. 193 at 1.) To preserve the Court’s and the parties’ resources, PAN had intended to wait for claim construction to file this motion,² but after the

¹ Contrary to Finjan’s misrepresentation, PAN did not “declare[] its intention to file its first motion to strike . . . before Finjan’s initial contentions were even due.” (Dkt. No. 199-2 (hereinafter “Opp.”) at 1.) PAN reminded Finjan of its past infringement contention failures and urged Finjan to comply. (Dkt. No. 104 at 23-25.)

² Claim construction may simplify or eliminate many of the issues raised in this motion. Despite knowing of PAN’s claim construction positions for over a year, Finjan has not addressed any of PAN’s constructions in its contentions. Should the Court adopt any of PAN’s claim

1 Court issued the scheduling order (Dkt. No. 194), PAN promptly conferred again with Finjan and
 2 filed this motion. (*See* Dkt. No. 196-6.)

3 II. ARGUMENTS

4 A. Finjan’s Contentions for the ’154 Patent Are Deficient

5 Finjan emphasizes that its contentions for the ’154 Patent are structured in four sections
 6 (Opp. at 3-4), but these sections do not disclose any cogent infringement theory. Finjan
 7 supposedly “introduces all infringement theories” in Section 1 (*id.* at 3), but Finjan discusses the
 8 operation of PAN’s products only at a high level, without specifying where the claim limitations
 9 can be found in PAN’s products. The discussion of the products’ operation is also not tied to the
 10 claim language, making it impossible to understand *how* PAN’s products allegedly infringe. The
 11 remaining sections are inconsistent with Section 1 or contain more descriptions of the operation
 12 of PAN’s products and source code without describing the required where and how.³

13 1. “input” and “content”

14 As with Finjan’s previously deficient contentions, its purportedly disclosed infringement
 15 theory in Section 1 does not “map [claim] components onto the features of the allegedly
 16 infringing products.” *Finjan, Inc. v. Check Point Software Techs., Inc.*, No. 18-CV-02621-WHO,
 17 2019 WL 955000, at *6 (N.D. Cal. Feb. 27, 2019). There is no dispute that the claim language
 18 requires “content” “received over a network” that “includ[es] a call to a first function” with “the
 19 call including an input.” (Dkt. No. 195-3 (“Mot.”) at 5; Opp. at 4.) Finjan’s attempt in its
 20 Opposition to explain how this limitation is met only highlights the deficiencies in its contentions.

21 Finjan’s Opposition confirms that in PAN’s products, when “content” is “received over a
 22 network,” the “content” does not “include[e] a call to a first function” with “the call including an
 23 input,” as required in the claims. At most, Finjan’s contentions show that “a call to a first
 24 function” with “the call including an input” is added to the “content” *after* it is “received over a

25 _____
 26 constructions (such as those that previous courts adopted or the Federal Circuit affirmed),
 27 Finjan’s contentions also are deficient for failing to address those constructions.

28 ³ PAN’s Motion identified representative deficiencies in Finjan’s infringement
 contentions. Finjan’s contentions against PAN’s Traps products are similarly deficient.

1 network,” which is contrary to what the claims require.

2 Despite repeatedly stating in its contentions that the accused “content” is “files, URLs,
3 and web content,” its Opposition concedes that this “content” does not meet the claims’
4 requirement that the “content” include “a call to a first function.” (Opp. at 5 (noting Finjan’s
5 theory is not that “files, URLs, and web content” include “a call to a first function”).) Finjan now
6 claims that “files, URLs, and web content” are just part of the accused “content.” (*Id.* at 4.)
7 Thus, the portions of Finjan’s contentions that rely on “files, URLs, and web content” alone as the
8 claims’ “content” fail to satisfy the required how because they cannot map to the claim language
9 that requires the “content” to include “a call to a first function.”

10 Having to disclaim its first “content” mapping, Finjan’s Opposition focuses on a second
11 mapping that artificially creates new “content” by combining “files, URLs, and web content” with
12 a PAN technology called SML files. (Opp. at 5 (“The accused content received over a network
13 including a call to a first function (substitute function), the call including an input is comprised of
14 content requested by the client computer, which is received over a network combined with SML
15 files received over a network at the NGFW.”).) But that mapping makes no sense as it is not how
16 PAN’s products operate and fails to clearly identify where in PAN’s products create the newly
17 combined “content.” Indeed, Finjan’s contentions concede that PAN’s products use SML files to
18 “parse and analyze the files, URLs, and web content.” (Dkt. No. 199-4 at 185.) PAN’s products
19 do not combine SML files with user data like “files, URLs, and web content.”

20 Even if, however, PAN’s products created this combined “content” (which Finjan does
21 not explain), this mapping still fails to follow the claim language and thus fails to satisfy the
22 Patent Local Rules. The claims require that, *when received over a network*, the “content” has “a
23 call to a first function” with an “input.” The SML files, according to Finjan, add the “call to a
24 first function” with an “input” that is missing from the “files, URLs, and web content” in its first
25 mapping. (Opp. at 5.) In other words, only after “files, URLs, and web content” are combined
26 with SML files is the “content” limitation met. (*Id.*) Thus, under this mapping, the “content”
27 including “files, URLs, and web content” does not have “a call to a first function” with an “input”
28 until *after it is received over a network* and combined with SML files in PAN’s products, which

1 is contrary to the claim language. (*See* Dkt. No. 199-4 at 13, 15-16.) Finjan has thus failed to
 2 identify how the claims’ “content” and “input” limitations are met in PAN’s products.

3 2. “first function” and “second function”

4 Finjan does not dispute that its Second Amended Contentions for the ’154 Patent are
 5 mostly the same as its original contentions, which Judge Hamilton struck. (*See* Mot. at 6.) Nor
 6 does it deny that, in Section 1 of its contentions, Finjan merely equates “first function” and
 7 “second function” with equally abstract “substitute function” and “original function” and fails to
 8 explain where and how the two limitations are met in PAN’s products. (*Id.*)

9 Finjan instead relies on Section 3 of its contentions, where it lists dozens of exemplary
 10 “first functions” and “second functions” without explaining how they meet the claim language.
 11 Finjan claims that the exemplary functions “tie to theories set forth elsewhere” but offers no
 12 examples. (Opp. at 5-6.) Finjan does not answer PAN’s question of how the alleged “first
 13 function” [REDACTED] meets the claim language. How is this function called with
 14 an “input”? And how is a call to this function included in the received “content”? Finjan does
 15 not say. Rather, Finjan argues that it met its burden under the Patent Local Rules because it
 16 “identifies source code in PAN’s products responsible for *generating* the ‘first function’ and
 17 ‘second function.’” (*Id.* at 5 (emphasis added).) But Finjan’s mapping again fails to follow the
 18 claim language. The claims say nothing of “generating” the first and second functions. Rather, a
 19 call to the “first function” must be included in the received “content.” Finjan thus again fails to
 20 map the claim language to PAN’s products.

21 Finjan’s mapping for “content” that combines “files, URLs, and web content” and “SML
 22 files” does not cure the deficiencies in Finjan’s infringement contentions. Finjan fails to describe
 23 how SML files include a call to any of the alleged “first functions” (such as
 24 [REDACTED]) or how they include any alleged “input.” In fact, Finjan fails to list
 25 any inputs to any of its “exemplary first functions.” (*See* Dkt. No. 199-4 at 295.)

26 Finjan’s Opposition confirms that it fails to explain *how* the claimed “first function” and
 27 “second function” share the same “input.” Despite Finjan’s emphasis on its identified exemplary
 28 functions, its cited excerpts merely use the terms “first function” and “second function” without

1 identifying where they are in PAN's products. (*See* Opp. at 7.) Moreover, the cited excerpts are
 2 divorced from the claim language. The excerpts discuss "transfer[ring] the input to the first
 3 function to the destination computer to be processed by the second function." (*Id.*) But that is
 4 not required by the claim language. Finjan does not describe how the "input" is included in "a
 5 call to a first function" and how the "second function" is invoked with the same "input."

6 3. "content processor" and "security computer"

7 Finjan does not dispute that its disclosed theory for "content processor" in Section 1 is
 8 conclusory and open-ended. (*See* Mot. at 8-9.) It instead requires that PAN decipher Finjan's
 9 infringement theory by studying Finjan's over 600 pages of contentions "in total." (Opp. at 7.)

10 Even when Finjan's contentions are read "in total," they fail to provide adequate notice of
 11 Finjan's theory for "content processor" and "security computer." Finjan cites pages 24-43
 12 (Section 2.1) of the '154 Chart, but the cited pages do not mention "content processor" at all, let
 13 alone identify where and how the claimed "content processor" is in PAN's products. (*See* Dkt.
 14 No. 199-4 at 24-43.) Finjan's cited source code section (*id.* at 177-184) does not disclose the
 15 where and how either. Finjan claims that it identifies source code that "implement[s] the claimed
 16 content processor" (Opp. at 7), but Finjan's cited pages do not state that the listed source code
 17 files implement the claimed "content processor." For example, on pages 178-179, Finjan lists
 18 source code files that implement "the modules that receive and process the network packets
 19 corresponding to files, URLs, and web content." (Dkt. No. 199-4 at 177-79). Are these modules
 20 the claimed "content processor" that processes "content"? Finjan leaves PAN guessing as
 21 Finjan's contentions have sometimes identified "files, URLs, and web content" as "input," while
 22 at other times as "content." *See supra* §II.A.1.

23 Finjan does not dispute that PAN has no modules named "pattern recognition modules,"
 24 which Finjan claims to be the "accused security computer." (Mot. at 8.) Contrary to Finjan's
 25 assertion (Opp. at 8), its contentions do not state that "pattern recognition modules" include
 26 PAN's "Malicious Signature Matching and Deterministic Finite Automata (DFA) Matching"
 27 modules, but rather that they "perform at least Malicious Signature Matching and Deterministic
 28 Finite Automata (DFA) Matching." (Dkt. No. 199-4 at 12.) Finjan points to no documentation

discussing Malicious Signature Matching and Deterministic Finite Automata (DFA) Matching. PAN therefore has no notice of what the “pattern recognition modules” are or where the “security computer” is in PAN’s products. Finjan’s citations to its contentions regarding WildFire are also filled with improper open-ended language. (*See, e.g.*, Opp. at 8 (“identifying WildFire, PAN-DB cloud, *and/or* Inline Machine Learning as security computers”) (emphasis added).)

B. Finjan’s Contentions for the ’408 Patent Are Deficient

1. “parser rules” and “analyzer rules”

Finjan’s Opposition confirms that it fails to identify *where* the “parser rules” and “analyzer rules” are in PAN’s products. Finjan claims that it “identifies SML files and DFA constructs as *including* the parser rules and analyzer rules” and cites portions of its contentions where it identifies “SML files and Deterministic Finite Automata (‘DFA’) constructs that *describe* parser and analyzer rules.” (*Id.* at 9-10 (emphasis added).)⁴ But where in the “SML files and Deterministic Finite Automata (‘DFA’) constructs” are the “parser rules” and “analyzer rules”? And even if SML files and DFA constructs were “parser rules” and “analyzer rules,” does Finjan contend that the “parser rules” and “analyze rules” are the same and they both are SML files and DFA constructs, or that “parser rules” are SML files and “analyzer rules” are DFA constructs? Finjan’s cited source code excerpt (*id.* at 9), which does not even mention “parser rules” or “analyzer rules,” does not provide the answer. Nor do Finjan’s contentions.

Finjan’s contentions regarding PAN’s WildFire product are similarly deficient. Finjan again fails to identify where the “parser rules” and “analyzer rules” are in WildFire, as Finjan’s cited pages demonstrate. (*See, e.g.*, Dkt. No. 199-5 at 144 (“programming language specific static and dynamic analyzers that *describe* parser and analyzer rules”) (emphasis added).) PAN also did not waive its arguments against Finjan’s WildFire contentions; Finjan acknowledges that

⁴ Finjan mischaracterizes its contentions. It quotes a paragraph cited by PAN and claims that the paragraph “makes clear that the ‘scanner . . .’ includes the parser rules and analyzer rules in the form of SML files and DFA construct[ions].” (Opp. at 9.) The quoted paragraph discusses “SML files and Deterministic Finite Automata (‘DFA’) constructs that *describe* parser and analyzer rules.” (*Id.* (emphasis added).) Finjan never explains where that description is or summarizes the description to explain where and how the parser and analyzer rules are present.

PAN moved to strike Finjan’s contentions in their entirety. (*See Opp.* at 10.) PAN’s Motion also made clear that PAN cited examples representative of the overall deficiencies of Finjan’s contentions. (*See Mot.* at 9.)

Finjan’s contentions do not explain how the “parser rules” and “analyzer rules” perform functionalities relating to the recited “patterns” and “tokens.” Finjan’s cited example merely repeats claim language. (*See id.* at 10 (“the parser and analyzer rules (e.g., SML files and DFA constructs) define certain patterns in terms of tokens and identify certain combinations of tokens and patterns as being indicators of potential exploits”).) Finjan does not identify where the “patterns” and “tokens” are in PAN’s products, let alone explain *how* the SML files and DFA constructs (supposed “parser and analyzer rules”) define or identify them. Finjan’s discussions of source code are irrelevant as Finjan “provides no explanation of how [the source code files or functions] map onto the claim language.” *Network Caching Tech. LLC v. Novell Inc.*, No. C-01-2079-VRW, 2002 WL 32126128, at *6 (N.D. Cal. Aug. 13, 2002) (finding infringement contentions insufficient).

2. “scanner”

Finjan’s Opposition confirms Finjan’s misreading of the claim language. The claim language is clear: a “scanner compris[es] parser rules and analyzer rules.” Finjan does not explain how specifying, calling, or relying on the rules meets the claim language. (*See Mot.* at 9.) Finjan, however, repeatedly maps the claimed “scanner” to the wrong claim language in arguing that it has identified the “scanner” as “content scanning engines,” “specific source code functions,” and “static and dynamic analyzers.” *Opp.* at 11-12 (“PAN’s NGFW content scanning engines (e.g., the claimed ‘scanner’) . . . *specify* parser rules and analyzer rules. . . . scanner source code *specifying* or *calling* the ‘parser rules’ and ‘analyzer rules. . . . WildFire has static analyzers that serve as scanners . . . scanning *relies* on parser and analyzer rules”) (emphasis added).) In Finjan’s quoted excerpts, the accused “scanner” also *specifies*, instead of *comprising*, the rules. (*Id.* at 11 (“The scanners instantiated by the [REDACTED] function *specify* parser rules and analyzer rules”), 12 (“The programming language specific static analyzer scanner *specify* parser rules and analyzer rules”) (emphasis added).) Finjan also cites to its purported

1 discussions of “pattern matching,” but Finjan again relies on the wrong claim language in these
 2 cited pages. (*Id.* at 11; Dkt. No. 199-5 at 113 (“content scanning engines *use* parser rules and
 3 analyzer rules”), 115 (“NGFWs use Inspection Processors and Discrete Finite Automata to
 4 *implement* analyzer rules.”) (emphasis added).) Finjan cannot satisfy the required how by relying
 5 on the wrong claim language.

6 Furthermore, although in a few places of its contentions, Finjan conclusorily claims that
 7 some components, such as NGFW’s “content scanning engines,” are the accused “scanners,”
 8 Finjan’s remaining contentions are untethered to these purportedly accused “scanners.” Finjan’s
 9 quoted excerpt on page 11 shows that. (*See, e.g.,* Opp. at 11.) Contrary to Finjan’s
 10 mischaracterization, the excerpt discusses the “scanners instantiated by the [REDACTED]
 11 function,” not “content scanning engines.” (*Id.*) The cited pages of the ’408 Chart also do not
 12 discuss “content scanning engine”; nowhere do the contentions state that “content scanning
 13 engines” are “scanners instantiated by the [REDACTED] function.” The discussions of some
 14 unidentified “scanners”—unrelated to “content scanning engines”—do not explain the required
 15 how for “content scanning engines.”

16 C. Finjan’s Contentions for the ’731 Patent Are Deficient

17 1. “file cache” and “security profile cache”

18 Finjan doubles down on its improper use of open-ended language. It quotes three
 19 paragraphs from its contentions with each paragraph containing similar open-ended language.
 20 (*See, e.g.,* Opp. at 13 (“a security profile cache (e.g., in a database, *such as* Local DB, Central
 21 DB, Virus Database, *or* in disk storage). . . . the file cache (e.g., a database, *such as* Local DB, *or*
 22 in disk storage/memory)”) (emphasis added).) That is not sufficient disclosure of an infringement
 23 theory. *Word to Info Inc. v. Google Inc.*, No. 15-CV-03486-WHO, 2016 WL 3648605, at *5 n.4
 24 (N.D. Cal. July 8, 2016) (warning plaintiff its improper “use of open-ended placeholder phrase
 25 like ‘such as’ and ‘for example’ [would] not enable it to rely on infringement theories not
 26 specifically articulated in its infringement contentions.”). Finjan’s vague contentions suggest that
 27 “file cache” and “security profile cache,” two different claim elements, can both be “Local DB.”
 28 (Mot. at 13.) By relying on open-ended language, Finjan seeks to be free to later claim that a

1 previously unidentified component was disclosed in its contentions. (*Id.* at 2.)

2 Finjan’s Opposition confirms PAN’s concern. Finjan now claims that its contentions
 3 “identify—by source code file—the ‘security profile cache.’” (Opp. at 13.) But Finjan does not
 4 say where exactly this identified “security profile cache” is in PAN’s products. Finjan’s cited
 5 pages only list source code files that purportedly “implement” “the operations for” “access and
 6 updating” or “indexing” the “security profile cache”; they do not identify a “security profile
 7 cache.” (See Dkt. No. 199-6 at 153-54, 156-57.) In addition to open-ended contentions, Finjan
 8 also claims that it can identify the claim limitations, such as “file cache,” “by functionality, e.g.,
 9 storing scanned files that are indexed by a file identifier.” (Opp. at 13.) But this is merely
 10 parroting claim language, not “identifying specifically where and how each limitation of each
 11 asserted claim is found within [the accused products].” Patent L.R. 3-1(c).

12 Finjan’s argument that its open-ended contentions are “based on the information PAN has
 13 made available thus far” (Opp. at 13) is meritless. Finjan must identify the accused “file cache”
 14 and “security profile cache” with specificity based on its available information and “delete
 15 references . . . to the infringement as ‘exemplary and not limiting’ or ‘by way of example and not
 16 limiting.’” *Finjan, Inc. v. FireEye, Inc.*, No. 13-cv-03133-SBA (JCS), Dkt. No. 134 (N.D. Cal.
 17 Oct. 16, 2017 (*see* Dkt No. 128-2)).

18 Although in a few places of its contentions, Finjan conclusorily states that “filecache1”
 19 and “filecache2” are the accused “file cache,” Finjan fails to describe the required how. (*See*
 20 Mot. at 12-13.) Finjan disputes this but cites excerpts where it merely discusses an unspecified
 21 “cache” or files being cached. (*See, e.g.*, Opp. at 14 (“PAN’s documentation shows that WildFire
 22 stores the sample (e.g., the scanned file) in a cache.”).) Finjan’s discussions of security profiles
 23 (*id.*) are similarly unrelated to “filecache1” and “filecache2.” Finjan cannot explain how
 24 “filecache1” and “filecache2” meet the claim language by pointing to some generic, unrelated
 25 discussions. *Shared Memory Graphics LLC v. Apple, Inc.*, 812 F. Supp. 2d 1022, 1025 (N.D.
 26 Cal. 2010) (the patentee “must map specific elements of Defendants’ alleged infringing products
 27 onto the Plaintiff’s claim construction.”).

28 Finjan’s contentions for “security profile cache” are divorced from the claim language.

1 The claims require the “security profile cache” to store “security profiles derived by the scanner,”
 2 which “comprises a list of computer commands.” (Mot. at 11.) Finjan does not dispute that its
 3 contentions regarding “security profile cache” do not mention “computer commands.” (*Id.* at 12.)
 4 It instead cites discussions of PAN’s products “scan[ning] results” or “storing scanned samples
 5 and results” (Opp. at 13), but that is not what the claim language requires.

6 2. “security policy cache”

7 PAN identified the deficiencies in Finjan’s contentions for “security policy cache” in
 8 PAN’s letter to Finjan (Dkt. No. 200-2 at 6), subsequent meet and confer, and PAN’s Motion
 9 (Mot. at 13-14). Finjan still cannot show that its contentions identify the required where and how.
 10 Finjan’s quoted language only invokes the term “security policy cache.” (*See* Opp. at 15
 11 (“security policy cache with the PAN-OS stores policies and rules . . .”).) But where is the
 12 accused “security policy cache” in PAN-OS? Finjan also claims that its contentions “identify the
 13 ‘security policy cache’ by source code files” (*id.*, citing Dkt. No. 199-6 at 169-171), but the cited
 14 pages show otherwise. Finjan merely list source code files that “implement[]” “the operations for
 15 accessing the security policy cache.” (*See* Dkt. No. 199-6 at 170.) But where is the “security
 16 policy cache” that is accessed by the operations implemented by the listed source code files?
 17 Because Finjan does not even specify where the accused “security policy cache” is in PAN’s
 18 products, it fails to explain how this limitation is met in PAN’s products. Finjan’s discussion of
 19 “restrictions” or citation of a YouTube video is irrelevant. *Uniloc USA, Inc. v. Apple Inc.*, No. C
 20 18-00360 WHA, 2018 WL 3219486, at *2 (N.D. Cal. July 2, 2018) (“The cited video, which
 21 explains how the accused feature operates at a high level, did not by itself specify where and how
 22 [the plaintiff] believes the infringing elements lie in the accused products.”).

23 III. CONCLUSION

24 Finjan has had three rounds of infringement contentions. Despite detailed explanations
 25 from PAN and guidance from the Court, Finjan still fails to identify where and how each of the
 26 claim limitations can be found in PAN’s products. The Court should strike Finjan’s Second
 27 Amended Contentions for the ’154, ’408, and ’731 Patents and dismiss Finjan’s claims of
 28 infringement of these patents with prejudice.

1
2 Dated: November 1, 2022

MORRISON & FOERSTER LLP

3
4 By: /s/ Diek O. Van Nort
5 Diek O. Van Nort

6 Attorneys for Defendant
7 PALO ALTO NETWORKS, INC.
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28